Formal Modelling and Verification of Probabilistic Resource Bounded Agents

Hoang Nga Nguyen[1†] and Abdur Rakib[2†]

[1]Department of Computer Science, Swanesa University, Fabian Way, Swansea, SA1 8EN, Wales, U.K..
[2]Institute for Future Transport and Cities, Coventry University, Priory Street, Coventry, CV1 5FB, West Midlands, U.K..

Contributing authors: h.n.nguyen@swansea.ac.uk; ad9812@coventry.ac.uk;
†These authors contributed equally to this work.

Abstract

Many problems in Multi-Agent Systems (MASs) research are formulated in terms of the abilities of a coalition of agents. Existing approaches to reasoning about coalitional ability are usually focused on games or transition systems, which are described in terms of states and actions. Such approaches however often neglect a key feature of multi-agent systems, namely that the actions of the agents require resources. In this paper, we describe a logic for reasoning about coalitional ability under resource constraints in the probabilistic setting. We extend resource-bounded alternating-time temporal logic (RB-ATL) [1] with probabilistic reasoning and provide a standard algorithm for the model-checking problem of the resulting logic Probabilistic resource-bounded ATL (pRB-ATL). We implement model-checking algorithms and present experimental results using simple multi-agent model-checking problems of increasing complexity.

Keywords: Logic of Resources, Alternating-time Temporal Logic, Probabilistic logic, Markov Decision Process, Multi-agent Systems

1 Introduction

An increasingly important field of AI is autonomous agents and multi-agent systems, where agents are entities that can interact with their environment or other agents in pursuit of their goals. In general, multi-agent systems research refers to software agents. However, the agents in a Multi-agent System (MAS) could also be for example humans or robots. A key feature of a MAS is that agents in the system are concurrent. The primary aims of such systems are modularity, scalability, flexibility, robustness and distributed computing [2]. In multi-agent systems, there are many problems which also require coalition formation among the agents and such problems can only be usefully analysed in terms of the combined abilities of groups of agents. For example, it may be that no single agent, a potential home buyer having some financial deposit constraint, has a strategy to reach a particular state, buying a home, on its own, but two agents, perhaps husband and wife, cooperating with each other are capable of achieving this outcome. Similarly, in the prisoners dilemma [3], a single prisoner cannot ensure the optimal outcome, while a coalition of two prisoners can. The Alternating-time Temporal Logic (ATL) [4] was introduced as a logical formalism for analysing the strategic abilities of coalitions with temporal winning conditions. The semantics of ATL is usually given by a transition system specification based on concurrent game structures. In ATL, various interesting properties of coalitions and strategies such as reachability and controllability can be formulated. One can then encode a system and verify its desired properties expressed in ATL using standard ATL model-checking tools, such as jMocha [5]. Coalition Logic (CL) [6] is another logical formalism similar to ATL, intended to describe the ability of groups of agents to achieve a goal in a strategic game. It can specify what a group of agents can (not) achieve through choices of their actions. In the literature, several variants of ATL and CL have been proposed (see e.g., [7–9]). These logics allow us to express many interesting properties of coalitions and strategies, such as "*a coalition of agents $A(\subseteq N$, the set of all agents) has a strategy to reach a state satisfying φ no matter what the other agents and/or environment ($N \setminus A$) in the system do*", where φ characterises, e.g., lifting a heavy weight by a group of robots A, saving a building from fire by a group of fire extinguisher robots A or simply a solution to a problem. In fact, these logics can be used to state various qualitative properties of real-world concurrent systems. However, analysing quantitative properties of systems, such as reliability and uncertainty, which can not be expressed trivially in the logics described above, is equally or even more important. Reliability is the probability that a system will perform its specified function over a given period of time under defined environmental conditions. For example, a reliability property could be *"if a fire is detected in a building, then the probability of it being put out and the building being saved by a coalition of robotic agents A within k time steps is at least p"*.

Many real-world systems, such as Internet of Things (IoT) and Cyber Physical Systems (CPS) are deeply rooted in activities of our daily living [10]. The multi-agent paradigm offers an excellent framework which can be used to model and implement such systems [11]. Such systems usually operate in unpredictable and/or uncertain environments [12, 13]. Their applications encompass many safety critical domains, and many such applications run in resource constrained devices and environments [14, 15]. These systems therefore often require rigorous analysis and verification to ensure their designs are correct [16]. In conventional verification via model checking, given a model of a system, and a specification, a model checker determines if the system satisfies the specification by returning a yes or no answer. However, when considering stochasticityin the environment, agents in the system should be formalised in a way so that they exhibit probabilistic behaviour. PRISM [17] is a tool for formal modelling and analysis of systems that exhibit random or probabilistic behaviour. A system model in PRISM can be created using its own modelling language similar to *reactive modules* [5], and the properties can be written in an appropriate property specification language, including PCTL [18], CSL [19], PLTL [20], PCTL* [20], and rPATL [21]. These are fundamentally probabilistic temporal logics. The logic rPATL allows to reason quantitatively about a system's use of resources and emphasises on expected reward related measures. In rPATL, we can express that a coalition of agents has a strategy which can ensure that either the probability of an event's occurrence or an expected reward measure meets some threshold. However, probabilistic resource-bounded properties such as:

- "*can coalition A have a strategy so that the probability to reach a state satisfying φ under the resource bound b is at least p?*";
- "*a coalition of agents A has a strategy to achieve a property φ with probability p provided they have resources b, but they cannot enforce φ under a tighter resource bound b'*";
- "*a coalition of agents A can maintain φ until ψ becomes true with probability p provided they have resources b*"; and
- "*if a property φ holds, then a coalition of agents A has a strategy to achieve a property ψ within n time steps with probability p provided they have resources b*"

can neither be expressed in rPATL nor in any other probabilistic temporal logics mentioned above in a straightforward way. In this paper, we propose a logic pRB-ATL for reasoning about coalitional ability under resource constraints in the probabilistic setting, which allows us to express such properties. This article is a revised and extended version of [22]. The main differences from [22] are a complete literature review, addition of the complete proofs of the lemmas and theorems, development of the model-checking toolkit, and modeling a more complex example system with experimental analysis and verification results.

The rest of this paper is organised as follows. In Section 2, we review related work and discuss how our proposed logic pRB-ATL differs from other logics suggested in the literature. In Section 3, we discuss the basic notions of probability distribution, and the underlying probabilistic formalisms of our logic such as Discrete-time Markov chains and Markov Decision Processes. In Section 4 we present the syntax and semantics of pRB-ATL. In Section 5, we give a model-checking algorithm for pRB-ATL. In Section 6, we outline an implementation of our model checking prototyping tool. In Section 7, we model, analyse, and present experimental results applying our techniques and tool. Finally, we conclude the paper in Section 8.

2 Related work

In this section, we outline recent important developments on ATL and its extensions considering conventional, resource-bounded, and probabilistic reasoning by discussing important features, such as (un)decidability results, expressiveness and model checking problems. In addition, we discuss a few other forms of logics that may deal with similar scenarios with uncertainty, such as epistemic logic, epistemic game theory, and fuzzy logic.

A large number of existing studies in the multi-agent coalition literature have formulated reasoning about the abilities of coalitions of agents in terms of games [6–8, 23]. The coalition logic basically generalises the notion of a strategic game, and its semantics is given in terms of state transition systems where each state has an associated strategic game. The logic ATL [4] was originally developed to reason about distributed processes in adversarial environments, and CL [6] is in fact can be conceived as the one-step fragment of ATL [7, 24]. That is, in CL, the outcome of a strategic game is realised in the next state, but in ATL, properties can be expressed holding in arbitrary future states. These logics allow us to express many interesting properties of coalitions and strategies, as mentioned previously, such as $\langle\langle A\rangle\rangle\varphi$, which states that coalition A has a strategy to reach a state satisfying φ. The exact semantics of the modalities of the coalition varies depending on whether or not the knowledge that each agent has about the current state of the game is complete (in modal logic it's attributed as complete/incomplete information), and whether agents can use past game state knowledge when deciding on their next move or not (in modal logic it's attributed as perfect/imperfect recall). It is shown in [4] that the model-checking problem for complete information is decidable in polynomial time, and it's believed to be undecidable for the incomplete information and perfect recall case.

Recently, there has been growing interest in formal models of resource-bounded agents [1, 25, 26]. In resource-bounded reasoning agent research work, the emphasis is on the behavior of agents constrained by fixed resource bounds. For example, the resource-bounded Alternating-time Temporal Logic (RB-ATL) [1] was developed for reasoning about coalitional ability under resource bounds. The logic RB-ATL allows us to express various resource-bounded properties, such as $\langle\langle A^b\rangle\rangle\varphi$ which expresses that coalition A has a strategy to reach a state satisfying φ under the resource bound b. The model checking problem for RB-ATL is decidable and if resource bounds are encoded in unary, the model-checking algorithm for RB-ATL runs in time polynomial in the size of the formula and the structure, and exponential in the number of resources. There also exist other works on extensions of temporal logics and logics of coalitional ability that are capable of expressing resource bounds [25, 26]. In [25] resource-bounded Tree Logics RTL and RTL* were introduced. The logic RTL*, which extends CTL* with quantifiers representing the cost of paths, can allow only to analyse single-agent systems. RTL is a fragment of RTL* in which each temporal operator is immediately preceded by a path quantifier. Fundamentally, in their proposed language the existential path quantifier $E\varphi$ of CTL has been replaced by $\langle\rho\rangle\varphi$, where ρ represents a set of available resources. Intuitively, the formula $\langle\rho\rangle\varphi$ states that there exists a computation feasible with the given resources ρ that satisfies φ. It has been shown that the model checking problem for RTL and some sub-classes of RTL* is decidable.

The Price resource-bounded ATL (PRB-ATL) logic proposed in [26] has introduced its model-checking problem and its syntax and semantics consider resource endowment of the whole system when evaluating a formula pertaining to a coalition of agents. In their model the resources are convertible to money and its amount is bounded. Example properties that can be expressed in PRB-ATL includes $\langle\langle A^{\$}\rangle\rangle\varphi$, which states that the coalition A has a strategy such that, no matter what the opponent agents do, φ can be achieved under the expenses \$. Similar to the RB-ATL, \$ can be ∞ in the general case. That is, the meaning of $\langle\langle A^{\$}\rangle\rangle\varphi$ when $\$ = \infty$ is the same as its counterpart in ATL. The model checking problem for PRB-ATL is decidable and its complexity similar to that of RB-ATL.

A large number of multi-agent application domains, such as IoT and CPS in general and disaster rescue and military operations in particular, require not only the reasoning about the team behavior of agents but also require that the agents and/or the environment may have random or unreliable behaviors. In such domains, the behaviour of an agent has to be described in terms of a distribution of probability over a set of possibilities. There has recently been increasing interest in developing logics with a probabilistic component and to link logical and probabilistic reasoning (see e.g., [21, 27–33]). These logics are essentially extensions of CTL or ATL which allow for probabilistic quantification of described properties. In general, probabilistic systems exhibit a combination of probabilistic and nondeterministic behaviour, and the semantics of the system models are defined in terms of probabilistic transition systems. For example, the semantics of the probabilistic ATL logics are defined over probabilistic extension of concurrent game structure [4], for which a commonly used underlying formalism is Markov Decision Processes (MDPs). Probabilistic model checking is also a well-established technique, and a well-known tool PRISM exists based on Markov chains (MCs) and MDPs probabilistic models [17]. In [27], PATL/PATL* logics have been developed by extending ATL and interpreting over the probabilistic concurrent game structures. Interesting properties that can be expressed in PATL includes $\langle\langle A\rangle\rangle[\varphi_p]\bowtie v$; it can be read as: a coalition A has a strategy such that for all strategies of agents not in A, the probability that the path formula φ is satisfied is $\bowtie v$ ($\bowtie \in \{\leq, <, >, \geq\}$). It was then further extended to develop the logics rPATL/rPATL* [21] for expressing quantitative properties of stochastic multiplayer games. The logics rPATL/rPATL* extend PATL/PATL* with operators that can enforce an expected reward $\bowtie v$. The logic rPATL* can express cumulative rewards given by the transition system. It is know that model checking problem for rPATL* is 2EXPTME-complete.

Similar to rPATL, strategies of the agents in our logic are randomised. An agent uses a randomised strategy by selecting a probability distribution over moves; and the move to be played is then chosen at random, according to the distribution. However, the reasoning problem considered in our work differs from rPATL in two important ways. First, and most importantly, properties in rPATL related to rewards are of statistical nature. They are expressed and computed as constraints on expected values for rewards. In contrast, resource-bounded properties in our pRB-ATL logic lie within the realm of crisp values and constraints; actions and strategies are allowed if and only if they satisfy the resource-bounded constraints. That is, using pRB-ATL, it is possible to ask whether a strategy (a sequence of actions) exists to achieve some goal with probability 0.99 if the agents start with e.g., 100 units of energy. Second, semantics for rPATL is based on turn-based systems while ours is based on concurrent systems. However, recently developed PRISM-games 3.0 [34] supports modelling concurrent games. We are aware that properties of resource-bounded systems can be verified by expressing them using rPATL. However, to encode a system using rPATL, we have to expand the model to incorporate the resource information into the states of the model. For different formulas with different resource bounds we have to then induce a new model to perform the model checking algorithms. Our proposed approach allows model checking algorithms to work directly with the original model. This opens up the possibility of future research including not only agents consume but also produce resources. In the resource production case, verifying resource bounded properties by encoding resource bounded system using rPATL is no longer feasible. Furthermore, we didn't mention anything about optimal strategies in our framework. Our aim is to check existence of a strategy (which may not be optimal), where the resources each agent is prepared to commit to a goal are bounded.

The approach proposed by [30] in developing probabilistic ATL logic relies on interpreted system semantics. The resulting logic PATL* essentially generalises the interpreted system [35] by adding probabilistic modality and explicit local actions taken by the agents. An example property that can be expressed in PATL* is $\langle\langle A\rangle\rangle^{\bowtie v}\varphi$ which expresses that coalition A has a strategy to enforce φ with a probability $\bowtie v$. However, since the semantics is based on incomplete information and synchronous perfect recall, model checking problem for PATL* is undecidable even for a single agent system.

In [28], another alternative semantics for a probabilistic logic PATL has been proposed using the notion of prediction denotation operator. In PATL the reasoning about probabilistic success studied over complete information games. The success of the strategy of a coalition is measured according to a probability measure describing the potential actions of the rest of the agents in the system. An example property that can be expressed in PATL is $\langle\langle A\rangle\rangle^{p}_{\omega}\varphi$ which expresses that coalition A has a strategy to enforce φ with probability p when agents not in A behave according to ω. The model checking problem for PATL with mixed strategies is bounded between Probabilistic polynomial time and PSPACE.

In [32], the model checking problem for probabilistic multi-agent systems is investigated using the probabilistic epistemic logic PETL, which can express both temporal and epistemic properties. The authors have shown that the model checking problem is in general undecidable; however, when reduced to the class of uniform memoryless schedulers, it becomes decidable. Finally, they presented a decision algorithm for the latter class, analysed its complexity (is in P^{NP} and NP-hard), and models from the International Probabilistic Planning Competitions were used to evaluate a prototype implementation. In [33], the authors introduced epistemic models considering DTMC as a stochastic modeling process. The proposed models extend interpreted systems in the probabilistic setting. It can model stochastic systems that are subject to stochastic phenomena. They have used PCTLK, a logic to reason about probabilistic and uncertain knowledge. The usability of the proposed framework has been demonstrated considering the Dining Cryptographers protocol. The PRISM implementation is compatible with the proposed modelling techniques and hence it can be used to model and verify desired system properties.

In order to include probability into coalition logics, the work by [36] presented two distinct methods, one based on the semantics of games with stochastic transitions and the other on games with stochastic failures. The first method captures the properties of games with stochastic transitions, while the second method captures the properties of games with stochastic failures. The two important technical findings for the games with stochastic failures are an incompleteness theorem that proves there is no strongly complete logical system in the language with the empty coalition and a strong completeness theorem for the logical system without the empty coalition.

In [37], the authors presented a probabilistic continuous-time linear logic (CLL), to reason about the probability distribution execution of continuous-time Markov chains (CTMCs). In CLL, multiphase timed until formulas are allowed and the semantics of the formulas focuses on relative time intervals, meaning that time can be reset just like in timed automata. The model-checking problem is reduced to a reachability problem of absolute time intervals.

In [38], Wang et al. proposed a new concept of probabilistic conformance for Cyber-Physical Systems (CSP). This idea is based on approximately equal satisfaction probability for a given (infinite) set of signal temporal logic (STL) formulas. They have presented a verification algorithm for the probabilistic compliance of grey-box CPS, described by probabilistic uncertain systems. Their proposed statistical verification method is based on a statistical test that can determine if two probability distributions are equal at any chosen level of confidence. It is shown that statistically confirming compliance is possible when the STL formula is monotonically parameterized, meaning that the satisfaction probability of the formula changes monotonically with the parameters.

The work presented in [39] takes a different approach, namely theorem proving, to specify a stochastic game with failure states and a logic of coalition power with modality that incorporates probability of non-failure. The authors provided the proofs of the results and a sound and complete axiomatization of the resulting logic for non-empty coalitions. The properties that can be expressed in the logic include a coalition C has a strategy that achieves φ in all non-failure states and is guaranteed to avoid failure states with probability at least p?. A coalition logic with an epistemic modality is presented in [40] as a fusion of STIT and temporal logics. The epistemic modality is used to express properties under uncertainty. A complete axiomatization is provided for the proposed (E)NCL logic and theorem proving can be used to reason about the resulting logic formulas. In [41], the authors proposed an approach considering fuzzy cooperative games and smart contracts to construct dynamic coalitions of independent robots (agents). The integrated model of game application with fuzzy coalitions and fuzzy smart contracts proposed in this research can be used to establish coalitions for humans and robots, as well as between multiple robots. It is proposed in this paper that smart contracts be used to store the rules for forming a coalition and rules for modifying the coalition's composition, which are defined using the fuzzy set theory. To illustrate the usability of the proposed approach, the authors show a use case based on precision farming.

We must say that after ATL [4] was introduced a remarkably rich literature has been developed. Here we have discussed only an overview of the works that are closely related to the topic of the paper. However, in all the approaches, at least in the probabilistic setting, the basic idea of agents acting in an environment according to a set of rules in the pursuit of goals does not take into account resources. In real life, many actions that an agent may perform to achieve a goal can only be accomplished in the availability of certain resources. Certain actions are not possible without sufficient resources, which will lead to a plan failure. To the best of our knowledge, there are no existing works in the literature that address probabilistic variants of ATL for modeling and verifying resource-bounded agents explicitly.

3 Background and Preliminaries

In this section, we discuss the basic notions that are used in the technical part of the proposed logic. Let Q be a finite set and $\mu : Q \to [0,1]$ be a *probability distribution* function over Q such that $\sum_{q \in Q} \mu(q) = 1$. We denote by $\mathcal{D}(Q)$ the set of all such distributions over Q. For a given $\mu \in \mathcal{D}(Q)$, $\mathsf{supp}(\mu) = \{q \in Q \mid \mu(q) > 0\}$ is called the *support* of μ. A *probability space* is a measure space with total measure 1. The standard notation of a *probability space* is a triple $(\Omega, \mathcal{F}, Pr)$, where Ω is a sample space which represents all possible outcomes, $\mathcal{F} \subseteq \mathcal{P}(\Omega)$ is a σ-algebra over Ω, i.e., it includes the empty subset and it is closed under countable unions and complement, and $Pr : \mathcal{F} \to [0,1]$ is a probability measure over $(\Omega, \mathcal{F})$. The interested reader is referred to [42] for a complete description relating to probability distributions and measures. We also denote the set of all finite, non-empty finite and infinite sequences of elements of Q by Q^*, Q^+ and Q^ω, respectively.

3.1 DTMC and MDP

Discrete-time Markov chains (DTMCs) are the simplest probabilistic models in which the systems evolve through discrete time steps.

Definition 1. *A DTMC is a tuple $M_c = (Q, q_0, \Pi, \pi, \delta)$, where Q is a set of states, $q_0 \in Q$ is the initial state, Π is a finite set of propositional variables, $\pi : Q \to \mathrm{wp}(\Pi)$ is a labelling function, and $\delta : Q \times Q \to [0,1]$ is a probability transition matrix such that $\sum_{q' \in S} \delta(q,q') = 1$ for all $q \in Q$.*

Here, $\delta(q,q')$ denotes the probability that the chain, whenever in state q, moves into next state q', and is referred to as a one-step transition probability. The square matrix $P = (\delta(q,q'))_{q,q' \in Q}$, is called the one-step transition matrix. Since when leaving state q the chain must move to one of the possible next states $q' \in Q$, each row sums to one.

Definition 2. *A path λ in a DTMC M_c is a sequence of states $q_0, q_1, q_2 \ldots$ such that $\delta(q_i, q_{i+1}) > 0$ for all $i \geq 0$. The i^{th} state in a path λ is denoted by $\lambda(i)$. The set of all finite paths starting from $q \in Q$ in the model M_c is denoted by $\Omega^+_{M_c,q}$, and the set* *of all infinite paths starting from q is denoted by $\Omega^{\omega}_{M_c,q}$. The prefix of the path λ of length n is $q_0, q_1, q_2 \dots q_n$.*

Definition 3. *A cylinder set C_λ is the set of infinite paths that have a common finite prefix λ of length n. Let $\Sigma_{\Omega_{M_c,q}}$ be the smallest σ-algebra generated by $\{C_\lambda \mid \lambda \in \Omega^{+}_{M_c,q}\}$. Then, we can define μ on the measurable space $(\Omega_{M_c,q}, \Sigma_{\Omega_{M_c,q}})$ as the unique probability measure such that:*

$$\mu(C_\lambda) = \prod_{i=0}^{|\lambda|-1} \delta(q_i, q_{i+1}).$$

Markov decision processes (MDPs), an extension to ordinary DTMCs, are widely used formalisms for modelling systems that exhibit both probabilistic and non-deterministic behaviour [29].

Definition 4. *An MDP is a tuple $M_d = (Q, q_0, \Pi, \pi, \mathcal{A}, \delta)$, where $\mathcal{A}$ is a set of actions, $\delta : Q \times \mathcal{A} \to \mathcal{D}(Q)$ is a (partial) probabilistic transition function, and all the other components are the same as their counterparts in a DTMC.*

The set of available actions at a state q is defined by $\mathcal{A}(q) = \{\alpha \in \mathcal{A} \mid \exists q' \cdot \delta(q, \alpha)(q') > 0\}$. Unlike DTMCs, in MDPs the transitions between states occur in two steps. Firstly, an action α is selected from a set of actions $\mathcal{A}(q)$ available at a given state q. Secondly, a successor state q' is chosen randomly, according to the probability distribution $\delta(q, \alpha)$. For a given state q and $\alpha \in \mathcal{A}(q)$, $\delta(q, \alpha) : Q \to [0, 1]$ is a function such that $\sum_{q' \in Q} \delta(q, \alpha)(q') = 1$.

Definition 5. *A path λ in an MDP M_d is an infinite alternating sequence of states and actions $\lambda = q_0 \xrightarrow{\alpha_0} q_1 \xrightarrow{\alpha_1} \dots \in (Q \times \mathcal{A})^\omega$ where $\alpha_i \in \mathcal{A}(q_i)$ and $\delta(q_i, \alpha_i)(q_{i+1}) > 0$ for all $i \geq 0$. A finite path $\lambda = q_0 \xrightarrow{\alpha_0} q_1 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_{n-1}} q_n$ is defined as usual as a prefix of an infinite path ending at a state q_n. The set of finite paths is denoted by $(Q \times \mathcal{A})^* Q$.*

Definition 6. *A strategy in an MDP M_d is a function $f : (Q \times \mathcal{A})^* Q \to \mathcal{D}(\mathcal{A})$ that assigns each finite path $\lambda = q_0 \xrightarrow{\alpha_0} q_1 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_{n-1}} q_n$ a probability distribution over the enabled actions $\mathcal{A}(q_n)$ such that $f(\lambda)(\alpha) > 0$ if $\alpha \in \mathcal{A}(q_n)$.*

We use a strategy to resolve the nondeterministic choices in an MDP. An MDP M_d's behaviour is entirely probabilistic under a specific strategy, resulting in a DTMC M_c. For a more detailed discussion we refer the interested reader to [43, p. 842-843, Definition 10.91. Scheduler].

4 Syntax and Semantics of pRB-ATL

In this section, we provide the syntax and semantics of pRB-ATL. Let us consider a multi-agent system consisting of a set $N = \{1, 2, \dots n\}$ of $n(\geq 1)$ concurrently executing agents. In order to reason about resources, we assume that the actions performed by the agents have costs. Let $R = \{res_1, res_2, \ldots, res_r\}$ be a finite set of $r \geq 1$ resources, such as money, energy, or anything else which may be required by an agent for performing an action. Without loss of generality, we assume that the cost of an action, for each of the resources, is a natural number. The set of resource bounds $\mathbb{B}$ over R is defined as $\mathbb{B} = (\mathbb{N} \cup \{\infty\})^r$, where $r = \mathsf{R}$. We denote by $\vec{0}$ the smallest resource bound $(0, \ldots, 0)$ and by $\vec{\infty}$ the greatest resource bound $(\infty, \ldots, \infty)$.

4.1 Syntax of pRB-ATL

Let Π be a finite set of atomic propositions and N be the set of agents. The syntax of $pRB\text{–}ATL$ is defined as follows:

$$\varphi := \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie v}[\psi]$$
$$\psi := \bigcirc\varphi \mid \varphi\, \mathcal{U}^k\, \varphi \mid \neg\psi$$

where $p \in \Pi$, $A \subseteq N$, $b \in \mathbb{B}$, $\bowtie \in \{<, \leq, \geq, >\}$, $v \in \mathbb{Q} \cap [0,1]$, and $k \in \mathbb{N} \cup \{\infty\}$.

The two temporal operators have the standard meaning, $\bigcirc$ for "next" and $\mathcal{U}^{\leq k}$ for "bounded until" if $k < \infty$ or "until" otherwise. When $k = \infty$, we shall simply write $\mathcal{U}$ instead of $\mathcal{U}^\infty$. Here, $\langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie v}[\bigcirc\varphi]$ means that a coalition A has a strategy to make sure that the next state satisfies φ under resource bound b with a probability in relation $\bowtie$ with constant v, regardless of the strategies of other players. The formula $\langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie v}[\varphi_1\, \mathcal{U}\, \varphi_2]$ means that A has a strategy to enforce φ_2 while maintaining the truth of φ_1, and the cost of this strategy is at most b with a probability in relation $\bowtie$ with constant v, regardless of the strategies of other players. Other temporal operators are defined as abbreviations in a standard way. Particularly, "eventually" is defined as $\Diamond\varphi \equiv \top\, \mathcal{U}\, \varphi$, and "always" as $\Box \equiv \neg\Diamond\neg\varphi$. Notice that these operators when $b = \vec{\infty}$ mean the same as their counterparts in ATL, i.e., the ATL operator $\langle\langle A \rangle\rangle$ corresponds to $\langle\langle A^{\vec{\infty}} \rangle\rangle$. Other classical abbreviations for $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined as usual.

4.2 Semantics of pRB-ATL

To interpret this language, we extend the definition of resource-bounded Concurrent Game Structures (RB-CGS)) [1] with probabilistic behaviours of agents. For consistency with [4], in what follows the terms 'agents' and 'players' and the terms 'actions' and 'moves' have been used interchangeably.

Definition 7. *A Probabilistic Resource Concurrent Game Structure (pRCGS) is a tuple* $S = (n, r, Q, \Pi, \pi, d, c, \delta)$ *where:*

- $n \geq 1$ *is the number of players (agents);*
- $r \geq 1$ *is the number of resources;*
- Q *is a non-empty finite set of states;*
- Π *is a finite set of propositional variables;*
- $\pi : \Pi \rightarrow \mathrm{wp}(Q)$ *is a function which assigns a subset of* Q *to each variable in* Π*;*

- $d : Q \times N \to \mathbb{N}_+$ *is a function which indicates the number of moves (actions) available at a state for each agent, where* $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$*;*
- $c : Q \times N \times \mathbb{N}_+ \to \mathbb{B}$ *is a partial function which maps a state* q*, an agent* a *and a move* $\alpha \leq d(q, a)$ *to a vector of integers where the integer in position* i *indicates consumption of resource* res_i *by the move* α*. We stipulate that* $c(q, a, 1) = \vec{0}$ *for any* $q \in Q$ *and* $a \in N$*;*
- $\delta : Q \times (N \to \mathbb{N}_+) \to \mathcal{D}(Q)$ *is a partial probabilistic transition function that for every* $q \in Q$ *and a joint move* m *gives the state resulting from executing* m *in* q*.*

Given a pRCGS $S = (n, r, Q, \Pi, \pi, d, c, \delta)$, we identify available moves at a state $q \in Q$ of an agent $i \in N$ by $1, \ldots, d(q, i)$; then $D_i(q) = \{1, \ldots, d(q, i)\}$ denotes the set of available moves; move 1 specifies idling which is always available with cost $\vec{0}$ by definition. Similar to ATL and RB-ATL, the zero-cost move 1 is required to avoid deadlock and, therefore, maintain totality.

A pRCGS is closely related to an MDP (Definition 4, Section 3.1), where abilities of individual agents and coalitions of agents are constrained by available resources in a non-trivial way. Given $A \subseteq N$, a joint move m of A is a function $m : A \to \mathbb{N}_+$. Given $q \in Q$, the set of available joint moves of A at q is denoted by $D_A(q) = \{m : A \to \mathbb{N}_+ \mid \forall a \in A : m(a) \in D_a(q)\}$. When $A = N$, we simply write $D(q)$ instead of $D_N(q)$ to denote the set of all joint actions for N at q. Given $q, q' \in Q$ and $m \in D(q)$, $\delta(q, m)(q')$ is the conditional probability of a transition from q to q' if every agent $i \in N$ performs $m(i)$. Then, q' is called a successor of q if $\exists m \in D(q)$ such that $q' \in \mathsf{supp}(\delta(q, m))$. To this end, pRCGS is different from RB-CGS in defining the transition function δ. While the δ of a RB-CGS [1] is deterministic, that of a pRCGS is a mapping to a distribution function over states and, hence, specifies non-determinism.

Example 1. *Let us consider the design of an autonomous firefighting system consisting of two firefighter agents* $N = \{1, 2\}$ *in a building. Each agent is equipped with two resources:* electricity *and* water*. Agents can perform three possible actions, namely,* sense*,* pump water *and* idle*. They can sense to detect if there is a fire in the building and pump water to stop the fire. Sensing the fire requires one unit of electricity, pumping water requires one unit of electricity and one unit of water, and idle costs nothing. This scenario is formalised by a pRCGS* S_{ff} *as depicted in Figure 1. Here,* $n = 2$*,* $r = 2$*,* $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$*, and* $\Pi = \{\text{low-burnt}, \text{medium-burnt}, \text{high-burnt}, \text{destroyed}\}$*. For convenience, the transition function* δ *is written in terms of labels on transitions. Each transition from a state* q_i *to a state* q_j *is annotated with one or more labels of the form* xy/z *where* xy *denotes the joint move,* x *is by agent 1's move and* y *is by agent 2's move performed at state* q_i*, and* z *denotes the probability of arriving at the next state* q_j*.*

At the initial state q_0*, each agent can either stay* idle *(*1*) or perform* sense *(*2*) action. Therefore, the possible joint moves at* q_0 *are* 11*,* 12*,* 21*, and* 22*. The states* q_1 *and* q_2 *represent circumstances in which the agents detect a fire either individually or as a coalition, respectively. The severity level of the fire is believed to be low in*

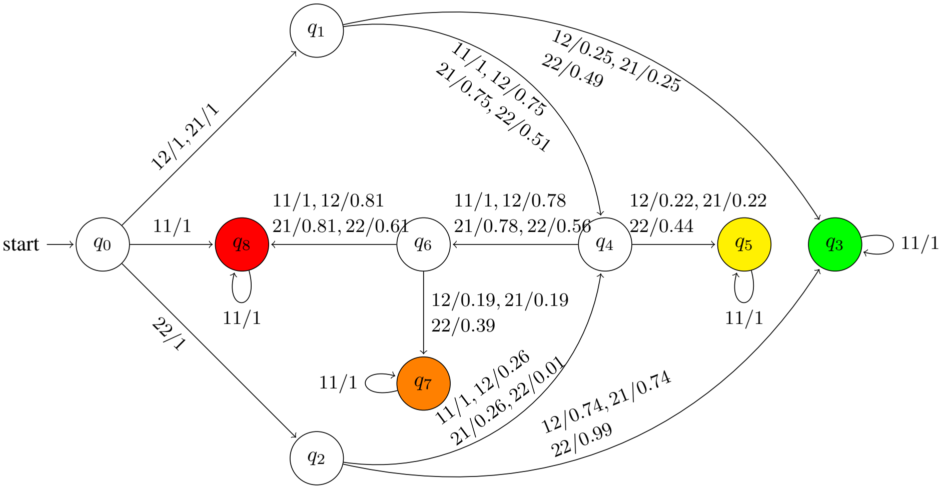

Fig. 1: pRCGS S_{ff} of the two firefighters. The proposition *"low-burnt"* is labelled on the "green" state q_3, *"medium-burnt"* is labelled on the "yellow" state q_5, *"high-burnt"* is labelled on the "orange" state q_7, and *"destroyed"* is labelled on the "red" state q_8.

these two states, that is, the building has just caught fire. At q_0, if both the agents stay idle and never sense to detect the fire, the system will enter state q_8 where the building can be burnt out completely. At q_1 (only one agent detected the fire) and q_2 (as a coalition both of them detected the fire), each agent can either stay idle *(*1*) or* pump water *(*2*). Thus, possible joint moves at each of these two states are* 11*,* 21*,* 12*, and* 22*. The system may then enter either q_3 or q_4 from both q_1 and q_2, depending on the actions performed by the agents. The "green" state q_3 reflects the low burnt condition of the building being saved shortly after the fire, and it is labelled with a proposition* "low-burnt". *However, the state q_4 implies an increased fire intensity level from low to medium severity. At q_4, agents can stay* idle *(*1*) or* pump water *(*2*). The system can then enter either q_5 or q_6 from q_4, depending on the actions performed by the agents. The "yellow" state q_5 reflects the medium burnt condition of the building being saved sometime after the fire, and it is labelled with a proposition* "medium-burnt". *Reaching to q_6 does, however, mean a further rise in fire intensity from medium to high severity. In the same way, at q_6 agents can stay* idle *(*1*) or* pump water *(*2*), and the system can then enter either q_7 or q_8 depending on the actions performed by the agents. The "orange" state q_7 reflects the high burnt condition of the building being saved long after the fire ignited, and it is labelled with a proposition* "high-burnt". *However, reaching to q_8 means that the building is completely destroyed and is labelled with a proposition* "destroyed".

If a fire does occur in the building, each agent autonomously decides to detect it or stay idle. When only one of the agents detects the fire, the chance of stopping it with "low-burnt" *condition is* 25% *if only one of them acts. However, the chance of stopping it increases to* 49% *if both of them act. The effectiveness of stopping the* *fire with* "low-burnt" *condition could be improved if both the agents detect the fire jointly. The chance of stopping it with* "low-burnt" *condition then would be* 74% *if only one of them acts, while it would be* 99% *if both of them act. If both the agents stay idle (i.e., neither of them detects the fire nor pumps the water to extinguish it), the building will be destroyed. Note that although the aim is to save the building with the* "low-burnt" *condition, it is not always possible. Thus, there are possibilities that the building fire intensity can be increased from low to medium, and eventually to high severity.*

To compare costs and resource bounds, we use the usual point wise vector comparison, that is, $(b_1, \dots, b_r) \leq (d_1, \dots, d_r)$ iff $b_i \leq d_i$ for $i \in \{1, \dots, r\}$ where $n \leq \infty$ for all $n \in \mathbb{N}$. We also use pointwise vector addition: $(b_1, \dots, b_r) + (d_1, \dots, d_r) = (b_1 + d_1, \dots, b_r + d_r)$ where $n + \infty = \infty$ for all $n \in \mathbb{N} \cup \{\infty\}$. Given $b \in \mathbb{B}$, $\mathbb{B}_{\leq b}$ denotes the set of bounds less than or equal to b without taking into account ∞ components, i.e., $\mathbb{B}_{\leq b} = \{b' \in \mathbb{B} \mid \forall i \in \{1, \dots, r\} : b'_i = b_i = \infty \vee b'_i \leq b_i < \infty\}$. Note that $\mid \mathbb{B}_{\leq b} \mid = \prod_{i \in \{1, \dots, r\}, b_i \neq \infty} (b_i + 1)$.

Given a joint move $m \in D_A(q)$, the cost of m is defined as:

$$cost(q, m) = \sum_{a \in A} c(q, a, m(a)).$$

That is, $cost(q, m)$ is the total cost of the actions performed by the agents in the coalition A.

Given a pRCGS S, we adopt the Definition 5 to define runs (computations). An infinite run is an infinite sequence $\lambda = q_0 \xrightarrow{m_0} q_1 \xrightarrow{m_1} \dots \in (Q \times D)^\omega$ where $m_i \in D(q_i)$ and q_{i+1} is a successor of q_i by m_i, i.e., $q_{i+1} \in \mathsf{supp}(\delta(q_i, m_i))$ for all $i \geq 0$. We denote the set of all infinite computations by $\Omega_S^\omega \subseteq (Q \times D)^\omega$. A finite computation is a finite prefix $\lambda = q_0 \xrightarrow{m_0} q_1 \xrightarrow{m_1} q_2 \dots \xrightarrow{m_{n-1}} q_n \in (Q \times D)^* Q$ of some infinite sequence in Ω_S. We denote the set of all finite computations by Ω_S^+ and the set of all finite and infinite computations by Ω_S, i.e., $\Omega_S = \Omega_S^+ \cup \Omega_S^\omega$. The length of a computation $\lambda \in \Omega_S$, denoted by $\mid \lambda \mid$, is defined as the number of transitions in λ. For a finite computation $\lambda = q_0 \xrightarrow{m_0} q_1 \xrightarrow{m_1} q_2 \dots \xrightarrow{m_{n-1}} q_n \in \Omega_S^+$, $\mid \lambda \mid = n$; for an infinite computation $\lambda = q_0 \xrightarrow{m_0} q_1 \xrightarrow{m_1} \dots \in \Omega_S^\omega$, $\mid \lambda \mid = \infty$. Given a computation $\lambda \in \Omega_S^+$, $\lambda(i) = q_i$ for all $i \in \{0, \dots, \mid \lambda \mid\}$; $\lambda(i, j) = q_i \dots q_j$ for all $i, j \in \{0, \dots, \mid \lambda \mid\}$ and $i \leq j$; $m_\lambda = m_0 m_1 \dots$ is the projection of moves in λ and $m_\lambda(i) = m_i$ for $i \in \{0, \dots, \mid \lambda \mid -1\}$. Note that $\lambda(\mid \lambda \mid)$ is the last state in λ. Finally, $\Omega_{S,q}^+ = \{\lambda \in \Omega_S^+ \mid \lambda(0) = q\}$ denotes the set of finite computations starting from $q \in Q$. Given a finite computation $\lambda \in \Omega_S^+$ and a coalition A, the cost of joint actions by A is defined as $cost_A(\lambda) = \sum_{i=0}^{|\lambda|-1} cost(\lambda(i), m_\lambda(i))$.

We adopt Definition 6 (Section 3.1) to define strategies as follows.

Definition 8. *Given a pRCGS S, a strategy of a player $a \in N$ is a mapping $f_a : \Omega_S^+ \rightarrow \mathcal{D}(\mathbb{N}_+)$ which associates each finite computation $\lambda \in \Omega_S^+$ to a distribution $\mu_a \in \mathcal{D}(D_a(\lambda(\mid \lambda \mid))$.*

Definition 9. *A strategy is called* memoryless *(or Markovian) if its choice of moves depends only on the current state, i.e., $f_a(\lambda) = f_a(\lambda(\mid \lambda \mid))$ for all $\lambda \in \Omega^+_S$. It is called* deterministic *if it always selects a move with probability 1, i.e., $f_a(\lambda)$ is a Dirac distribution.*

Definition 10. *Given a pRCGS S, a coalition strategy $F_A : A \to (\Omega^+_S \to \mathcal{D}(\mathbb{N}_+))$ is a function which associates each player a in A with a strategy.*

Given a coalition strategy F_A, we show that each finite computation $\lambda \in \Omega^+_S$ gives rise to a distribution $\mu^{F_A}_\lambda \in \mathcal{D}(D_A(\lambda(\mid \lambda \mid)))$ over joint actions $m \in D_A(\lambda(\mid \lambda \mid))$ where $\mu^{F_A}_\lambda(m) = \prod_{a \in A} f_a(\lambda)(m(a))$ and $f_a = F_A(a)$ for all $a \in A$.

Lemma 1. *Given a finite computation $\lambda \in \Omega^+_S$ and a coalition strategy F_A, $\mu^{F_A}_\lambda$ is a distribution over $D_A(\lambda(\mid \lambda \mid))$.*

Proof Let $q = \lambda(\mid \lambda \mid)$. It is trivial that $\mu^{F_A}_\lambda(m) \in [0, 1]$ for all $m \in D_A(q)$. It remains to show that $\sum_{m \in D_A(q)} \mu^{F_A}_\lambda \ (m) = 1$. It is done by induction on the cardinality of A. When $\mid A \mid= 1$, it is trivial. Assume that $\mid A \mid> 1$, let b be some agent in A and D_X denote $D_X(q)$, we have:

$$\begin{aligned}
&\textstyle\sum_{m \in D_A} \mu^{F_A}_\lambda(m) = \sum_{m \in D_A} \prod_{a \in A} F_A(a)(\lambda)(m(a)) \\
&= \textstyle\sum_{i \in D_b} \sum_{m' \in D_{A \setminus \{b\}}} F_A(b)(\lambda)(i) \times \prod_{a \in A \setminus \{b\}} F_A(a)(\lambda)(m'(a)) \\
&= \textstyle\sum_{i \in D_b} (F_A(b)(\lambda)(i) \times \sum_{m' \in D_{A \setminus \{b\}}} \prod_{a \in A \setminus \{b\}} F_A(a)(\lambda)(m'(a))) \\
&= \textstyle(\sum_{i \in D_b} F_A(b)(\lambda)(i)) \times (\sum_{m' \in D_{A \setminus \{b\}}} \prod_{a \in A \setminus \{b\}} F_A(a)(\lambda)(m'(a))) \\
&= \textstyle 1 \times \sum_{m' \in D_{A \setminus \{b\}}} \mu^{F_{A \setminus \{b\}}}_\lambda(m') \overset{ih}{=} 1 \times 1 = 1
\end{aligned}$$

Given two coalition strategies F_A and F_B of two disjoint coalitions A and B, i.e., $A \cap B = \emptyset$, their union is also a coalition strategy, denoted by $F_A \cup F_B$, for $A \cup B$.

Definition 11. *Given a bound $b \in \mathbb{B}$ and a strategy F_A, F_A is b-bounded iff for all $\lambda \in \Omega^+_S$ such that $cost_A(\lambda) \leq b$, it holds that $\mathsf{supp}(\mu^{F_A}_\lambda) \subseteq \{m \in D_A(\lambda(\mid \lambda \mid)) \mid cost_A(\lambda(\mid \lambda \mid), m) \leq b - cost_A(\lambda)\}$.*

In other words, all executions of a b-bounded strategy cost at most b resources. In order to reason about probabilistic behaviour of S, we need to determine the probability that certain computations are taken. To do this, we construct for each state $q \in Q$, a *probability space* over the set of infinite computations $\Omega^\omega_{S,q}$ starting from q. The basis of the construction is the probability of individual finite computations induced by the transition probability function δ. Given a state $q_0 \in Q$, we can determine the probability of every finite computation $\lambda = q_0 \xrightarrow{m_0} q_1 \xrightarrow{m_1} q_2 \ldots \xrightarrow{m_{n-1}} q_n \in \Omega^+_{S,q_0}$ consistent with F_A as follows:

$$\mathsf{Pr}^{F_A}_{S,q_0}(\lambda) = \prod_{i=0}^{n-1} \mu^{F_A}_{\lambda(0,i)}(m_i) \cdot \delta(q_i, m_i)(q_{i+1}).$$

If $\mid \lambda \mid= 1$, $\mathsf{Pr}^{F_A}_{S,q_0}(\lambda) = 1$ as the above product is empty.

For each finite computation $\lambda \in \Omega^{+}_{S}$, we can then define a cylinder set C_λ that consists of all infinite computations prefixed by λ. Given an initial state $q \in Q$, it is then standard [42, 44] to define a measurable space over $\Omega^{\omega}_{S,q}$, infinite runs of S from q, as $(\Omega^{\omega}_{S,q}, \mathcal{F}_{S,q})$ where $\mathcal{F}_{S,q} \subseteq \mathrm{wp}(\Omega^{\omega}_{S,q})$ is the least σ-algebra on $\Omega^{\omega}_{S,q}$ generated by the family of all cylinder sets C_λ where $\lambda \in \Omega^{+}_{S,q}$. Given a strategy F_N, a strategy for all players in the game, the behaviour of S is fully probabilistic. It then gives rise to a probability measure $(\Omega^{\omega}_{S,q}, \mathcal{F}_{S,q}, \mathsf{Pr}^{F_N}_{S,q})$ where $\mathsf{Pr}^{F_N}_{S,q} : \mathcal{F}_{S,q} \to [0,1]$ uniquely extends $\mathsf{Pr}^{F_N}_{S,q} : \Omega^{+}_{S,q} \to [0,1]$ such that $\mathsf{Pr}^{F_N}_{S,q}(C_\lambda) = \mathsf{Pr}^{F_N}_{S,q}(\lambda)$ for all finite computations $\lambda \in \Omega^{+}_{S,q}$.

4.3 Truth definition for pRB-ATL

Given a pRCGS $S = (n, r, Q, \Pi, \pi, d, c, \delta)$, the truth definition for pRB-ATL is given inductively as follows:

- $S, q \models \top$;
- $S, q \models p$ iff $q \in \pi(p)$;
- $S, q \models \neg\varphi$ iff $S, q \not\models \varphi$;
- $S, q \models \varphi_1 \vee \varphi_2$ iff $S, q \models \varphi_1$ or $S, q \models \varphi_2$;
- $S, q \models \langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie v}[\psi]$ iff $\exists$ b-bounded F_A such that $\forall F_{N\setminus A}$, $\mathsf{Pr}^{F_A \cup F_{N\setminus A}}_{S,q}(\{\lambda \in \Omega_{S,q} \mid S, \lambda \models \psi\}) \bowtie v$;
- $S, \lambda \models \bigcirc\varphi$ iff $S, \lambda(1) \models \varphi$;
- $S, \lambda \models \varphi_1 \,\mathcal{U}^k\, \varphi_2$ iff $\exists i \in \mathbb{N}$ such that $i \leq k$, $\forall j < i : S, \lambda(j) \models \varphi_1$ and $S, \lambda(i) \models \varphi_2$;
- $S, \lambda \models \neg\psi$ iff $S, \lambda \not\models \psi$.

This definition is a combination of pATL and RB-ATL. In particular, the case of $\langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie z}[\psi]$ requires the existence of a strategy F_A which must be b-bounded while there is no restriction on the strategies of the remaining players $\bar{A} = N \setminus A$.

From the truth definition, the following result directly inherits the complement rule for probability where $\geq^{-1} \equiv \leq$, $>^{-1} \equiv <$, $\leq^{-1} \equiv \geq$ and $<^{-1} \equiv >$:

Lemma 2. $\forall S, q : S, q \models \langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie v}[\psi] \Leftrightarrow S, q \models \langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie^{-1} 1-v}[\neg\psi]$.

Proof

S,q $\models \langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie v}[\psi]$ iff $\exists$ b-bounded F_A such that
$\forall F_{\bar{A}}, \mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\{\lambda \in \Omega_{S,q} \mid S, \lambda \models \psi\}) \bowtie v$
iff $\exists$ b-bounded F_A such that
$\forall F_{\bar{A}}, \mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\Omega_{S,q} \setminus \{\lambda \in \Omega_{S,q} \mid S, \lambda \models \psi\}) \bowtie^{-1} 1 - v$
iff $\exists$ b-bounded F_A such that
$\forall F_{\bar{A}}, \mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\{\lambda \in \Omega_{S,q} \mid S, \lambda \not\models \psi\}) \bowtie^{-1} 1 - v$
iff $\exists$ b-bounded F_A such that
$\forall F_{\bar{A}}, \mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\{\lambda \in \Omega_{S,q} \mid S, \lambda \models \neg\psi\}) \bowtie^{-1} 1 - v$
iff $S, q \models \langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie^{-1} 1-v}[\neg\psi]$

Example 2. *Let us continue with the running Example 1. Consider a question (property of the system):* "Can agent 1 from q_0, equipped with 2 units of electricity and 1 unit of water, make sure that the building is at least 49% low-burnt safe?". *This means to check if* $\varphi_{\{1\}} = \langle\langle \{1\}^{(2,1)} \rangle\rangle \mathsf{P}_{\geq 0.49} \Diamond$low-burnt *is true at* q_0. *Unfortunately, there is no such strategy for agent*

1, *i.e.,* $S_{ff}, q_0 \not\models \varphi_{\{1\}}$. *Consider another question (property of the system):* "Can agents 1 and 2 jointly from q_0, equipped with 4 units of electricity and 2 units of water, make sure that the building is at least 74% low-burnt safe?". *Similar to the previous question, we need to check if* $\varphi_{\{1,2\}} = \langle\!\langle \{1,2\}^{(4,2)} \rangle\!\rangle \mathsf{P}_{\geq 0.74} \Diamond \text{low-burnt}$ *is true at* q_0. *This is true, for example, when employing a strategy where both the agents perform sensing at* q_0 *and at least one of them pumping the water at* q_2. *In fact, this strategy can guarantee the low-burnt safety of the building by up to* 99%. *Hence,* $S_{ff}, q_0 \models \varphi_{\{1,2\}}$.

5 Model checking

In probabilistic model checking, the most elementary class of properties for probabilistic models is reachability. Given a state $q \in Q$, the probabilistic reachability problem computes the probability to reach some state in a specified target set of states in the model. That is, the basic reachability question is: *"can we reach a given target state from a given initial state with some given probability* v*?"*. More formally, given a state $q \in Q$, and a set of target states $T \subseteq Q$, the reachability probability is the measure of paths starting in q and containing a state from T, i.e., $Pr(\{\lambda \in \Omega_{M,q} \mid \lambda(i) \in T \text{ for some } i \in \mathbb{N}\})$. The property of probabilistic reachability actually refers to the minimum or maximum probability. In practice, many model checking problems can be reduced to reachability problem; therefore, it is considered as one of the most fundamental properties in probabilistic model checking. For an in-depth discussion on this topic, we refer the interested reader to [29].

Here, we present an algorithm for the model checking problem of pRB-ATL. In particular, given a pRCGS $S = (n, r, Q, \Pi, \pi, d, c, \delta)$ and a pRB-ATL formula φ, the algorithm produces the set of states $Sat(\varphi)$ of S that satisfy φ, i.e., $Sat(\varphi) = \{q \in Q \mid S, q \models \varphi\}$. Similar to ATL and its descendants, the algorithm generally processes φ recursively by computing the set of states satisfying sub-formulae of φ before combining them to produce $Sat(\varphi)$. For the propositional cases, the algorithm can be summarised as follows:

$$Sat(\top) = S, \qquad Sat(\varphi_1 \vee \varphi_2) = Sat(\varphi_1) \cup Sat(\varphi_2),$$
$$Sat(\neg\varphi) = S \setminus Sat(\varphi), \qquad Sat(p) = \{q \in Q \mid q \in \pi(p)\}.$$

Let us focus on the last cases $Sat(\langle\!\langle A^b \rangle\!\rangle \mathsf{P}_{\bowtie v}[\psi])$ where $\psi = \bigcirc\varphi_1$ and $\psi = \varphi_1 \, \mathcal{U}^k \, \varphi_2$ with $k \in \mathbb{N} \cup \{\infty\}$. Notice that the case $Sat(\langle\!\langle A^b \rangle\!\rangle \mathsf{P}_{\bowtie v}[\neg\psi])$ can be reduced to $Sat(\langle\!\langle A^b \rangle\!\rangle \mathsf{P}_{\bowtie^{-1} 1-v}[\psi])$ due to Lemma 2. Instead of following the semantics definition, i.e., determining the existence of a b-strategy for A to achieve a certain probability v from a state s, we compute the min and max values over all possible b-strategies for A. In particular:

$$\mathsf{Pr}^{\max}_{S,q}(A^b, \psi) = \sup_{b\text{-bounded } F_A} \inf_{F_{\bar{A}}} \mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\psi)$$
$$\mathsf{Pr}^{\min}_{S,q}(A^b, \psi) = \inf_{b\text{-bounded } F_A} \sup_{F_{\bar{A}}} \mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\psi),$$

where $\mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\psi)$ is a shorthand for

$$\mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\{\lambda \in \Omega_{S,q} \mid S, \lambda \models \psi\}).$$

Then, computing states satisfying $\langle\!\langle A^b \rangle\!\rangle \mathsf{P}_{\bowtie v}[\psi]$ is reduced to comparing these min/max values with v as follows:

$$Sat(\langle\!\langle A^b \rangle\!\rangle \mathsf{P}_{\rhd v}[\psi]) = \{q \in Q \mid \mathsf{Pr}^{\max}_{S,q}(A^b, \psi) \rhd v\} \tag{1}$$
$$Sat(\langle\!\langle A^b \rangle\!\rangle \mathsf{P}_{\lhd v}[\psi]) = \{q \in Q \mid \mathsf{Pr}^{\min}_{S,q}(A^b, \psi) \lhd v\}, \tag{2}$$

where $\lhd \in \{<, \leq\}$ and $\rhd \in \{>, \geq\}$. To this end, we can formulate a simple recursive algorithm, as presented in Algorithm 1, to compute the set of states satisfying a formula φ. It still remains to show how to compute $\mathsf{Pr}^{\max}_{S,q}(A^b, \psi)$ and $\mathsf{Pr}^{\min}_{S,q}(A^b, \psi)$. Based on the structure of ψ, these values can be computed according to the following three cases:

Algorithm 1 Computing $Sat(\varphi)$

```
function $Sat(\phi)$
    case $\varphi = \top$
        return $S$
    case $\varphi = \varphi_1 \vee \varphi_2$
        return $Sat(\varphi_1) \cup Sat(\varphi_2)$
    case $\varphi = \neg\varphi'$
        return $S \setminus Sat(\varphi')$
    case $\varphi = p$
        return $\{q \in Q \mid q \in \pi(p)\}$
    case $\varphi = \langle\langle A^b \rangle\rangle \mathsf{P}_{\rhd v}[\psi]$
        return $\{q \in Q \mid \mathsf{Pr}^{\max}_{S,q}(A^b, \psi) \rhd v\}$
    case $\varphi = \langle\langle A^b \rangle\rangle \mathsf{P}_{\lhd v}[\psi]$
        return $\{q \in Q \mid \mathsf{Pr}^{\min}_{S,q}(A^b, \psi) \lhd v\}$
end function
```

Case (a): $\psi = \bigcirc\varphi_1$. Assume that $Sat(\varphi_1)$ has been computed. Then the maximal probability to arrive at a state in $Sat(\varphi_1)$ is obtained by players in A selecting an allowed move (costing at most b) to maximise the probability while players outside A, i.e., in $\bar{A}$, select an arbitrary move to minimise it. Conversely, the minimal probability is obtained by players in A selecting an allowed move to minimise it while those outside select one to maximise it. Therefore, we have:

$$\mathsf{Pr}^{\max}_{S,q}(A^b, \psi) = \max_{\substack{m \in D_A(q) \\ cost(q,m) \leq b}} \min_{m' \in D_{\bar{A}}(q)} \sum_{t \in Sat(\varphi_1)} \delta(q, m \cup m')(t)$$

$$\mathsf{Pr}^{\min}_{S,q}(A^b, \psi) = \min_{\substack{m \in D_A(q) \\ cost(q,m) \leq b}} \max_{m' \in D_{\bar{A}}(q)} \sum_{t \in Sat(\varphi_1)} \delta(q, m \cup m')(t).$$

Case (b): $\psi = \varphi_1 \, \mathcal{U}^{\leq k} \, \varphi_2$. Assume that $Sat(\varphi_1)$ and $Sat(\varphi_2)$ are computed. For convenience, we denote $\mathsf{Pr}^{\max}_{S,q}(A^b, \varphi_1 \, \mathcal{U}^k \, \varphi_2)$ and $\mathsf{Pr}^{\min}_{S,q}(A^b, \varphi_1 \, \mathcal{U}^k \, \varphi_2)$ by $X^b_{q,k}$ and $Y^b_{q,k}$, respectively. Then, there are three trivial sub-cases:

- $q \in Sat(\varphi_2)$: any computation from q satisfies ψ, hence $X^b_{q,k} = Y^b_{q,k} = 1$.
- $q \notin Sat(\varphi_1) \cup Sat(\varphi_2)$: any computation from q does not satisfy ψ, hence $X^b_{q,k} = Y^b_{q,k} = 0$.
- $q \notin Sat(\varphi_2)$ and $k = 0$: any computation from q does not satisfy ψ before 0 transition, hence $X^b_{q,k} = Y^b_{q,k} = 0$.

Otherwise, players in A try to choose an allowed move m from q with cost at most b that maximises the probability to arrive at a state that can satisfy ψ with the remaining resource

$b' = b - cost(q, m)$ and within $k' = k - 1$ transitions. Formally, this can be defined as follows:

$$X^b_{q,k} = \max_{\substack{m \in D_A(q) \\ cost(q,m) \leq b}} \min_{m' \in D_{\bar{A}}(q)} \sum_{t \in Q} \delta(q, m \cup m')(t) \cdot X^{b-cost(q,m)}_{t,k-1}$$

$$Y^b_{q,k} = \min_{\substack{m \in D_A(q) \\ cost(q,m) \leq b}} \max_{m' \in D_{\bar{A}}(q)} \sum_{t \in Q} \delta(q, m \cup m')(t) \cdot Y^{b-cost(q,m)}_{t,k-1}.$$

Overall, one can form two linear equation systems with variables $X^b_{q,k}$ and $Y^b_{q,k}$, respectively, for each k. They can be solved by direct methods such as Gaussian elimination or iterative methods such as Jacobi and Gauss-Seidel [44]. In general, iterative methods suit the two linear equation systems best. It should not iterate more than $k + 1$ times as $X^b_{q,0}$ and $Y^b_{q,0}$ saturate to either 0 or 1 regardless of b and q by definition.

Case (c): $\psi = \varphi_1 \,\mathcal{U}\, \varphi_2$. Assume that $Sat(\varphi_1)$ and $Sat(\varphi_2)$ are computed. Again for convenience, we denote $\mathsf{Pr}^{\max}_{S,q}(A^b, \varphi_1 \,\mathcal{U}\, \varphi_2)$ and $\mathsf{Pr}^{\min}_{S,q}(A^b, \varphi_1 \,\mathcal{U}\, \varphi_2)$ by X^b_q and Y^b_q, respectively. Then, X^b_q can be computed by iterating the computation of $X^b_{q,k}$ up to a large enough k such that $\mid X^b_{q,k} - X^b_{q,k+1} \mid$ is less than some threshold ϵ. This is based on the fact that $X^b_{q,k}$ is a non-decreasing sequence that converges to $X^b_{q,k}$ [45].

In the following, we show the termination and the correctness of Algorithm 1.

Theorem 1. *Given a pRCGS S and a pRB-ATL formula φ, (i) $Sat(\varphi)$ terminates and (ii) $q \in Sat(\varphi)$ iff $S, q \models \varphi$.*

Proof Intuitively, termination is straightforward due to the fact that recursive calls within $Sat(\varphi)$ are always applied to strictly sub-formulas of φ. Let us prove (i) and (ii) by induction on the structure of φ.

Base case:

- If $\varphi = \top$, then $Sat(\top) = S$. That means (i) holds immediately and (ii) follows directly from the truth definition.
- If $\varphi = p \in \Pi$, then $Sat(p) = \pi(p)$ by the Sat definition. That also means (i) holds immediately and for (ii):
  $q \in Sat(\varphi)$ iff $q \in \pi(p)$ by Sat definition
  iff $S, q \models p$ by the truth definition.

Induction case:

- If $\varphi = \neg\varphi_1$, then $Sat(\varphi) = S \setminus Sat(\varphi_1)$ by the Sat definition. By the induction hypothesis, $Sat(\varphi_1)$ terminates, therefore, (i) holds. For (ii):
  $q \in Sat(\varphi)$ iff $q \in S \setminus Sat(\varphi_1)$ by the Sat definition
  iff $q \notin Sat(\varphi_1)$
  iff $S, q \not\models \varphi_1$ by the induction hypothesis
  iff $S, q \models \neg\varphi_1$ by the truth definition.
- If $\varphi = \varphi_1 \vee \varphi_2$, then $Sat(\varphi) = Sat(\varphi_1) \cup Sat(\varphi_2)$ by the Sat definition. By the induction hypothesis, $Sat(\varphi_1)$ and $Sat(\varphi_2)$ terminate, therefore, (i) holds. For (ii):
  $q \in Sat(\varphi)$ iff $q \in Sat(\varphi_1) \cup Sat(\varphi_2)$ by the Sat definition
  iff $q \in Sat(\varphi_1)$ or $q \in Sat(\varphi_2)$
  iff $S, q \models \varphi_1$ or $S, q \models \varphi_2$ by the induction hypothesis
  iff $S, q \models \varphi_1 \vee \varphi_2$ by the truth definition.
- If $\varphi = \langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie v}[\psi]$, then $Sat(\varphi) = \{q \in Q \mid \mathsf{Pr}^{mm}_{S,q}(A^b, \psi) \bowtie v\}$ by the Sat definition where $mm = \max$ if $\bowtie \in \{\geq, >\}$ and $mm = min$ if $\bowtie \in \{\leq, <\}$.
  To prove that (i) holds, we must show $\mathsf{Pr}^{mm}_{S,q}(A^b, \psi)$ terminates.

- If $\psi = \bigcirc\varphi_1$, by the Case (a), the calculation of $\mathsf{Pr}^{mm}_{S,q}(A^b, \psi)$ terminates due to the fact that $Sat(\varphi_1)$ terminates by the induction hypothesis, and $D_A(q)$, $D_{\bar{A}}(q)$, $Sat(\varphi_1)$ are all finite.
- If $\psi = \varphi_1 U^{\leq k}\varphi_2$, by the cases (b) and (c), the calculation of $\mathsf{Pr}^{mm}_{S,q}(A^b, \psi)$ terminates due to the fact that $Sat(\varphi_1)$ and $Sat(\varphi_2)$ terminate by the induction hypothesis, $D_A(q)$, $D_{\bar{A}}(q)$, $Sat(\varphi_1)$, $Sat(\varphi_2)$ are all finite, and the solution of the corresponding linear equation systems also terminates.

For (ii): If $mm = \max$,

$q \in Sat(\varphi)$ iff $\mathsf{Pr}^{\max}_{S,q}(A^b, \psi) \triangleright v$ by the Sat definition
iff $\sup_{b\text{-bounded } F_A} \inf_{F_{\bar{A}}} \mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\psi) \triangleright v$
iff $\exists$ b-bounded $F_A, \forall F_{\bar{A}}$: $\mathsf{Pr}^{F_A \cup F_{\bar{A}}}_{S,q}(\psi) \triangleright v$
since the model is finite, sup and inf are turned
into max and min, respectively,
in Cases (a), (b) and (c)
iff $S, q \models \varphi$ by the truth definition.

When $mm = \min$, the proof is symmetric to the one above, hence, omitted here.

□

Assuming that the natural numbers occurring in a pRB-ATL formula φ are encoded in unary, we have the following result.

Theorem 2. *The upper bound of the time complexity for $Sat(\varphi)$ is $O(\mid \varphi \mid^{3r+1} \cdot \mid S \mid^3)$.*

Proof φ has at most $O(\mid \varphi \mid)$ sub-formulae. The case (c) is the most computationally complex. In this case, b is bounded by $O(\mid \varphi \mid^r)$. Therefore, the number of variables for each iteration, also that of equations, in each corresponding linear equation system is bounded by $O(\mid \varphi \mid^r \cdot \mid S \mid)$. It is well-known that the time complexity of solving such a linear equation system is at most $O(n^3)$ [46], where n is the number of equations. Therefore, the upper bound complexity of computing $Sat(\varphi)$ is $O(\mid \varphi \mid \cdot(\mid \varphi \mid^r \cdot \mid S \mid)^3) = O(\mid \varphi \mid^{3r+1} \cdot \mid S \mid^3)$. □

Furthermore, the lower bound is given by that of the ATL, i.e., linear to the size of the input model and the input formula.

6 Tool Implementation

We have developed a prototype probabilistic model checking tool for resource-bounded stochastic multiplayer games based on the techniques proposed in this paper [1]. The tool is implemented in Python. It takes two input, a pRCGS model and a pRB-ATL formula. The model input is then interpreted by a parser into an instance of the class `Model`. Similarly, the formula is interpreted by an another parser into an instance of the class `Formula`. This formula instance may recursively include further formula instances of the sub formulas of the input formula. Finally, the implementation of the model checking procedure $Sat(\varphi)$, introduced in Section 5, is executed to compute the set of states of the input model satisfying the input formula. The whole described process is depicted in Figure 2.

The class `Formula` has five sub-classes corresponding to the five cases of state formulas φ defined in Section 4.1. For the last case $\langle\langle A^b \rangle\rangle \mathsf{P}_{\bowtie v}[\psi]$, an auxiliary class, named `PathFormula`, is introduced to represent path formulae (next, until and negation). Similar to

[1] https://github.com/ngasoft/mcPRBATL

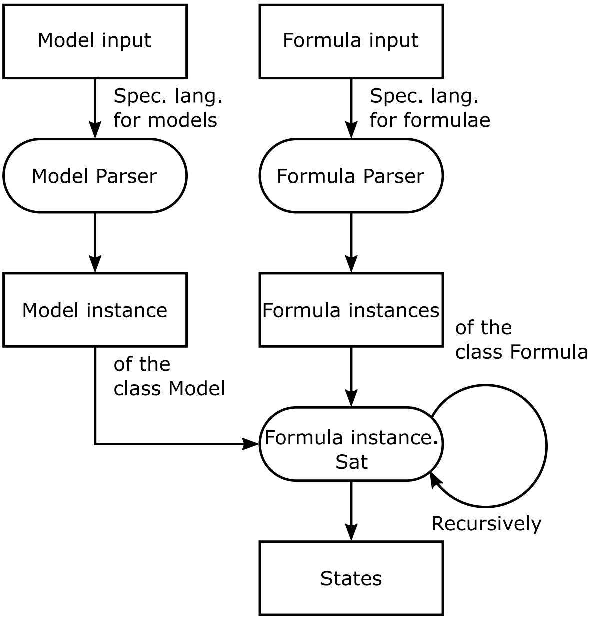

Fig. 2: The implementation of pRB-ATL model checking process.

the class `Formula`, `PathFormula` has three sub-classes corresponding to the three cases of the path formulae ψ.

The two parsers have been implemented based on Antlr4 [47]. They interpret pRCGS models and pRB-ATL formulae in a specification language, respectively, defined as close to the syntaxes of pRCGS and pRB-ATL as possible. The syntax of the specification language for pRCGS models is as follow:

```
'Structure' NAME '=' '{' agents ','  resources ','
  gstates ',' propositions ',' labellings ',' availables
  ',' costings ',' transitions  '}'
```

where `agents` is a positive number indicating the number of agents in the model, `resources` is a number indicating the number of resources. The remainders describe the set of their corresponding elements. In particular, sets such as `gstates` are defined by the syntax diagram depicted in Figure 3 where `gstate` are simply names of states. The systax diagram

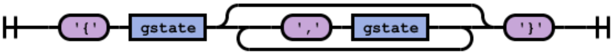

Fig. 3: The syntax diagram for sets in a pRCGS model.

for `propositions` is similar. Functions such as `availables` are defined as sets of mappings. Each mapping follows the syntax diagram in Figure 4 from a pair consisting of a state and a number (identifying an agent) to the number of available actions. The syntaxes are sim-

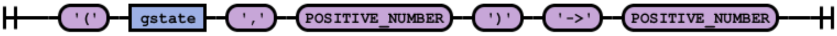

Fig. 4: The syntax diagram for mapping in a pRCGS model.

ilar for `labellings`, `costings`, `transitions` where elements of `labellings` are mapping to a set, `costings` to a cost (a tuple of numbers) and `transitions` to a state distribution. A state distribution is simply a set of pairs consisting of a state and a natural number. The probability of a state in a distribution is the division of its corresponding number by the sum of all the numbers in the distribution. The syntax diagram for such a pair is depicted in Figure 5.

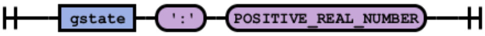

Fig. 5: The syntax diagram for an element of a state distribution.

The underlying implementation technique of the model checking procedure $Sat(\varphi)$ is an explicit-state model checking. The procedure of $Sat(\varphi)$ is implemented by a member method, named `sat`, of the class `Formula`. This method is overridden for each of the five sub-classes corresponding to the five cases of $Sat(\varphi)$ described in Section 5. The method `sat` takes only one parameter, an instance of the input model, and recursively calls the same method of instances representing the sub-formulae of φ.

7 Experimental results

Let us illustrate the use of the pRB-ATL and quantitatively verify the system presented in Example 1 via our model checking tool. We generalise the properties described in Example 2 as follows *"Can a coalition from q_0, equipped with e units of electricity and w units of water, make sure that the building is at least v low-burnt safe?"*. This is formalised in pRB-ATL by $\varphi_A = \langle\!\langle A^{(e,w)} \rangle\!\rangle \mathsf{P}_{\geq v} \Diamond \textit{low-burnt}$. To this end, we need to check whether $S_{ff}, q_0 \models \varphi_A$. As mentioned in the previous section, this can be reduced to determine the maximal probability:

$$\mathsf{Pr}^{\max}_{S_{ff},q_0}(A^{(e,w)}, \Diamond \textit{low-burnt}) = X^{(e,w)}_{q_0}.$$

Intuitively, the best strategy for one agent to help the building to be saved with low-burnt is to sense and then to pump the water. In total, this costs two units of electricity and one unit of water. Similarly, while cooperating, the best strategy for both the agents would be to choose their best strategies concurrently, which will cost together four units of electricity and two units of water. Therefore, for $A = \{1\}$, we consider the resources bounded by $(e, w) \leq (2, 1)$ and for $A = \{1, 2\}$ those are bounded by $(e, w) \leq (4, 2)$. For each case of A and (e, w), the model checking results are summarised in Table 1 for $A = \{1\}$ and in Table 2 for $A = \{1, 2\}$. In particular, Table 1 shows that any resource bound less than $(2, 1)$ is not helpful for agent 1 as it has no strategy to make sure that the building is safe with low-burnt. In the best case, with resource bound $(2, 1)$, the only vital strategy is to sense the fire and then pump the water. In this case, since agent 2 is not required to cooperate, the worst case is to end up in q_1 from q_0 where the chance to arrive at q_3, the low-burnt safe state, is at least 25%. That is, choosing the following actions: 21 in q_0, 21 in q_1 and 11 in q_3. Note that any resource bound greater than $(2, 1)$ will also not increase the chance of making sure the building is low-burnt safe with a higher probability. Similarly, Table 2 shows that any resource bound less than $(2, 1)$ will not be enough for both the agents while cooperating. However, the chance of making the building safe with low-burnt increases to 74% as more and more resources are given. This is because from q_0 both the agents can force the arrival at q_2 instead of q_1. Eventually, the maximal chance of making the building safe with low-burnt reaches 99% as both the agents have enough resources to follow the same best strategy. That is, choosing the following actions: 22 in q_0, 22 in q_2 and 11 in q_3.

Property type	Model checking result				Time (s)
$\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{1\}^{(e,w)}, \Diamond \textit{low-burnt})$	w/e	0	1	2	0.08
	0	0.0	0.0	0.0	
	1	0.0	0.0	0.25	
$\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{1\}^{(e,w)}, \Diamond \textit{medium-burnt})$	w/e	0	1	2	0.08
	0	0.0	0.0	0.0	
	1	0.0	0.0	0.0572	
$\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{1\}^{(e,w)}, \Diamond \textit{high-burnt})$	w/e	0	1	2	0.09
	0	0.0	0.0	0.0	
	1	0.0	0.0	0.0385	

Table 1: $\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{1\}^{(e,w)}, \Diamond\varphi)$

As we mentioned earlier, if a fire occurs in the building, there are possibilities that the building fire intensity can be increased from low (at q_1 or q_2) to medium (at q_4), and eventually to high (at q_6) severity. If the fire intensity reaches from low to medium, then for a single agent with resource bound $(2, 1)$ the maximum probability for which the building can be saved with *medium-burnt* is 5.7%, choosing the following actions: 22 in q_0, 12 in q_2, 21 in q_4, and 11 in q_5. Since agent 2 is not required to cooperate, it will try to minimize the probability. Basically, there are two paths from q_0 to q_5 where agent 1 performs action 2 at q_0, namely $q_0 \xrightarrow{22} q_2 \xrightarrow{12} q_4 \xrightarrow{21} q_5$ with probability 0.057 and $q_0 \xrightarrow{21} q_1 \xrightarrow{12} q_4 \xrightarrow{21} q_5$ with probability 0.165, and obviously the first path will be chosen. An interesting point to note here is that increasing resource bound from $(2, 1)$ to $(3, 2)$ for agent 1 will not increase the probability of saving the building anymore with medium-burnt. For example, with resource bound $(3, 2)$ if agent 1 senses the fire and then pumps water twice, then agent 2, being uncooperative and its objective is the opposite, will sense the fire and pump the water once, and ultimately will lead through a path ending up with probability 0.002. That is, choosing the following actions: 22 in q_0, 22 in q_2, 21 in q_4, and 11 in q_5. Tables 1 and 2 demonstrate the self-explanatory results for the coalition, as well as other situations such as high-burnt.

7.1 Modelling and analysis of the firefighting system

To demonstrate the usability and applicability of our proposed techniques and tool, and to evaluate their performance, we present results from a more generalised version of the example

Property type	Model checking result						Time (s)
$\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{1,2\}^{(e,w)}, \Diamond\textit{low-burnt})$	w/e	0	1	2	3	4	0.09
	0	0.0	0.0	0.0	0.0	0.0	
	1	0.0	0.0	0.25	0.74	0.74	
	2	0.0	0.0	0.25	0.74	0.99	
$\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{1,2\}^{(e,w)}, \Diamond\textit{medium-burnt})$	w/e	0	1	2	3	4	0.09
	0	0.0	0.0	0.0	0.0	0.0	
	1	0.0	0.0	0.22	0.22	0.22	
	2	0.0	0.0	0.22	0.44	0.44	
$\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{1,2\}^{(e,w)}, \Diamond\textit{high-burnt})$	w/e	0	1	2	3	4	0.10
	0	0.0	0.0	0.0	0.0	0.0	
	1	0.0	0.0	0.19	0.19	0.19	
	2	0.0	0.0	0.19	0.39	0.39	

Table 2: $\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{1,2\}^{(e,w)}, \Diamond\varphi)$

system discussed above. We modelled the system as a pRCGS with $n \in \{2, 3, \ldots, 13\}$ players, and we check the same pRB-ATL properties as above considering varying $A \subseteq N$ to be coalitions of different sizes, $A \in \{\{1\}, \{1, 2\}, \ldots, N\}$.

We increase or decrease the problem size by parameterizing the number of agents, and use a script to generate a complex system encoding. The script takes input as the number of agents and resources, and creates a system model in which the agents generate their alternative actions. The number of agents also parameterises the probability distribution of transitions. Note that, for a simple representation, Figure 1 shows some transitions from state q_i to state q_j by collapsing several possible combinations of actions. These combinations of actions will increase as the number of agents in the system increases. For experimental consistency, we parameterise how the probabilities are assigned to the transitions. In the case of two agents, we have two possible states in which a low fire can be detected, namely q_1 and q_2. The possible moves from each of these two states are $11, 12, 21$ and 22. Moves 12 and 21, where only one agent acts, have the same probability. Thus, if we leave the move 11 with probability 1 which takes the system from q_1 to q_4 (or q_2 to q_4), then each of the remaining $12, 21$ and 22 moves will require appropriate probability distribution over next states. In this case, q_3 in which the building is saved with *low-burnt* can be reached from q_1 and q_2 via four different transitions: (1) $q_1 \xrightarrow{12/21} q_3$, (2) $q_1 \xrightarrow{22} q_3$, (3) $q_2 \xrightarrow{12/21} q_3$, and (4) $q_2 \xrightarrow{22} q_3$. We assign probabilities to these transitions in an increasing order. The first transition has the lowest chance of saving the building with a low burnt condition, so we assign the lowest probability to it, while we assign the highest probability to the last transition, which has the highest chance of saving the building. We assume that with the low burnt condition the maximum chance of a building can be saved is 99%. So, we divide 0.99 by the number of these distinct moves, i.e., we assign $p = 0.99/4$ $(= 0.2475 \approx 0.25)$ to the move (1) and $1 - p$ to the corresponding transition from q_1 to q_4. We then increase it by p each time probability for the next transition

	move	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9
q_0	111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
	112/121/211	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	122/212/221	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	222	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
q_1	111	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	112/121/211	0.0	0.0	0.0	0.0	0.11	0.89	0.0	0.0	0.0	0.0
	122/212/221	0.0	0.0	0.0	0.0	0.22	0.78	0.0	0.0	0.0	0.0
	222	0.0	0.0	0.0	0.0	0.33	0.67	0.0	0.0	0.0	0.0
q_2	111	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	112/121/211	0.0	0.0	0.0	0.0	0.44	0.56	0.0	0.0	0.0	0.0
	122/212/221	0.0	0.0	0.0	0.0	0.55	0.45	0.0	0.0	0.0	0.0
	222	0.0	0.0	0.0	0.0	0.66	0.34	0.0	0.0	0.0	0.0
q_3	111	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	112/121/211	0.0	0.0	0.0	0.0	0.77	0.23	0.0	0.0	0.0	0.0
	122/212/221	0.0	0.0	0.0	0.0	0.88	0.12	0.0	0.0	0.0	0.0
	222	0.0	0.0	0.0	0.0	0.99	0.01	0.0	0.0	0.0	0.0
q_4	111	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
q_5	111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
	112/121/211	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.9	0.0	0.0
	122/212/221	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.8	0.0	0.0
	222	0.0	0.0	0.0	0.0	0.0	0.0	0.29	0.71	0.0	0.0
q_6	111	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
q_7	111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
	112/121/211	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.09	0.91
	122/212/221	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.17	0.83
	222	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.26	0.74
q_8	111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
q_9	111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Table 3: Transition table for three agents

is assigned, and the final value of p which is basically 0.99 is assigned to the move (4) (and $1 - p$ to the corresponding transition from q_2 to q_4). In the case of three agents, we have three possible states in which a low fire can be detected based on the following actions performed at q_0: $(211/1, 121/1, 112/1)$, $(221/1, 122/1, 212/1)$, and $(222/1)$. From each of these states, there will be three possible moves for which we need to assign appropriate probabilities. Thus, the lowest probability would be $p = 0.99/9$, then we increase it by p each time probability for the next transition is assigned. Table 3 demonstrates three agents' transitions. In general, in a model, the probability distribution to the low fire states would be assigned using $p = 0.99/(n * n)$, where n in the number of agents. The similar type of probability distribution is used for the medium and high fire states. Also, q_{n+1}, q_{n+3}, q_{n+5}, and q_{n+6} represent the "green", "yellow", "orange", and "red" states of the n agent transition diagram, respectively.

We conducted an extensive set of experiments, however, Table 4 presents the most significant results. Our purpose here is not to provide a detailed analysis of the time and space required to model different classes of pRB-ATL formulae, but simply to give an indication of the scalability and effectiveness of our algorithms and their implementation. Table 4 shows experiments run on an $Intel(R)\ Core(TM)\ i5$-$6500@3.20\ GHz$ using $8\ GB\ RAM$. It includes model statistics, number of agents, states and transitions, and the times to construct a pRCGS S_{ff} model and to verify a property of the form $\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{A\}^{(e,w)}, \Diamond\varphi)$. The results only include two extreme coalitions, such as the single agent coalition $\{1\}$ and the coalition N of all the agents in the system. However, our experiments suggest that when there is an increase in the number of players in a coalition the verification time increases.

pRCGS statistics					Time	
#Agents	Coalition	Resource	#States	#Transitions	Construction (Sec.)	Verification (Sec.)
2	$\{1\}$	$(1,2)$	$2,621$	$5,508$	0.03	0.08
	N	$(2,4)$	$6,509$	$12,366$	0.04	0.09
3	$\{1\}$	$(1,2)$	$3,272$	$13,080$	0.06	0.11
	N	$(3,6)$	$15,152$	$50,400$	0.07	0.28
4	$\{1\}$	$(1,2)$	$3,995$	$31,812$	0.13	0.23
	N	$(4,8)$	$29,735$	$185,262$	0.13	0.83
5	$\{1\}$	$(1,2)$	$4,790$	$77,328$	0.30	0.52
	N	$(5,10)$	$52,310$	$631,728$	0.28	2.64
6	$\{1\}$	$(1,2)$	$5,657$	$186,108$	0.65	1.19
	N	$(6,12)$	$85,217$	$2,028,078$	0.65	8.87
7	$\{1\}$	$(1,2)$	$6,596$	$442,244$	1.14	2.75
	N	$(7,14)$	$131,084$	$6,196,512$	1.53	27.85
8	$\{1\}$	$(1,2)$	$7,607$	$1,038,420$	3.54	6.80
	N	$(8,16)$	$192,827$	$18,173,790$	3.60	82.24
9	$\{1\}$	$(1,2)$	$8,690$	$2,410,176$	8.11	15.61
	N	$(9,18)$	$273,650$	$51,516,864$	8.12	249.54
10	$\{1\}$	$(1,2)$	$9,845$	$5,537,580$	18.64	36.66
	N	$(10,20)$	$377,045$	$141,911,070$	18.65	734.52
11	$\{1\}$	$(1,2)$	$11,072$	$12,609,432$	48.13	95.78
	N	$(11,22)$	$506,792$	$381,521,232$	49.19	2089.57
12	$\{1\}$	$(1,2)$	$12,371$	$28,485,636$	97.29	203.03
	N	$(12,24)$	$666,959$	$1,004,507,694$	98.02	5828.11
13	$\{1\}$	$(1,2)$	$13,742$	$63,899,760$	217.09	450.10
	N	$(13,26)$	$861,902$	$2,597,327,760$	217.34	14417.33

Table 4: Performance statistics. Property type $\mathsf{Pr}^{\max}_{S_{ff},q_0}(\{A\}^{(e,w)}, \Diamond \textit{low-burnt})$

Figures 6 and 7 depict experimental performance comparison considering the two extreme coalitions. This illustrates how the size of the coalition and number of agents in the system affects the performance of our model-checking algorithms. Note that our results are not directly comparable to the existing probabilistic model checking results. The use of resource bounds make our models much more complex and increases the non-determinism that exists within the transition relations of systems.

8 Conclusions and future work

In this paper, we have designed and developed a framework for automatic verification of systems with both resource limitations and probabilistic behaviour. We proposed a new temporal logic pRB-ATL for reasoning about coalitional abilities of systems under resource constraints that exhibit both probabilistic and non-deterministic behaviour. The novelty of our approach lies in complex logical combinations that tackles the problem of more comprehensive resource-bounded probabilistic multi-agent system specification and verification. To model multi-agent systems where the actions of agents consume resources, we have modified probabilistic strategy logics in two ways. Firstly, we added resource annotations to the actions in the transition system. For each individual action and each resource type, we have specified how many units of this resource type the action consumes. Secondly, we have extended the logical language so that we can express properties related to resources and probability. The model-checking problem for standard strategy logics is a special case of the model-checking problem for the corresponding resource logics. We have investigated how much harder does the model-checking

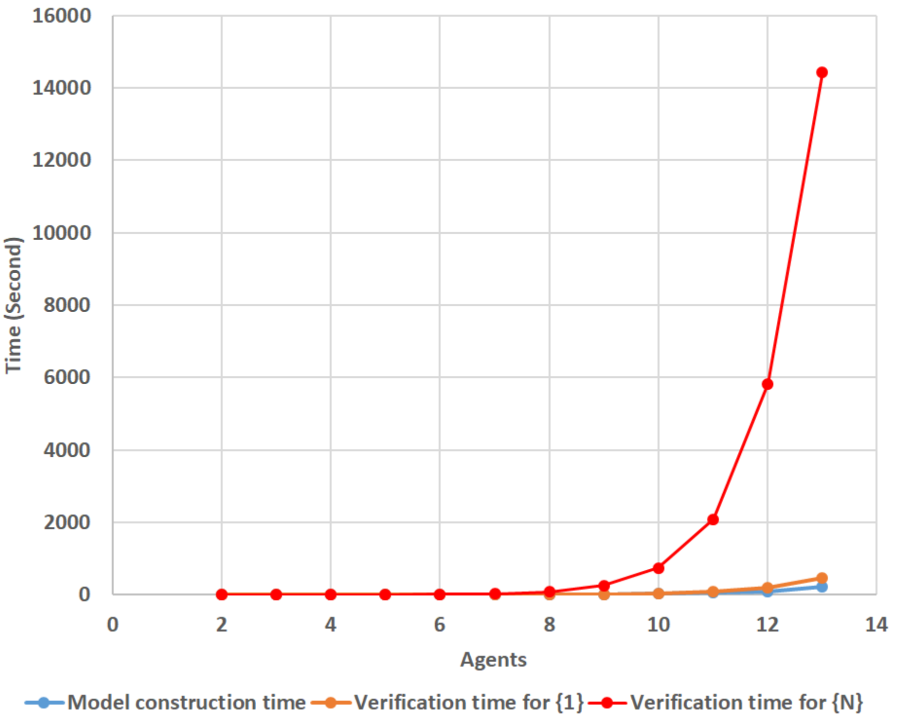

Fig. 6: Model construction and verification time

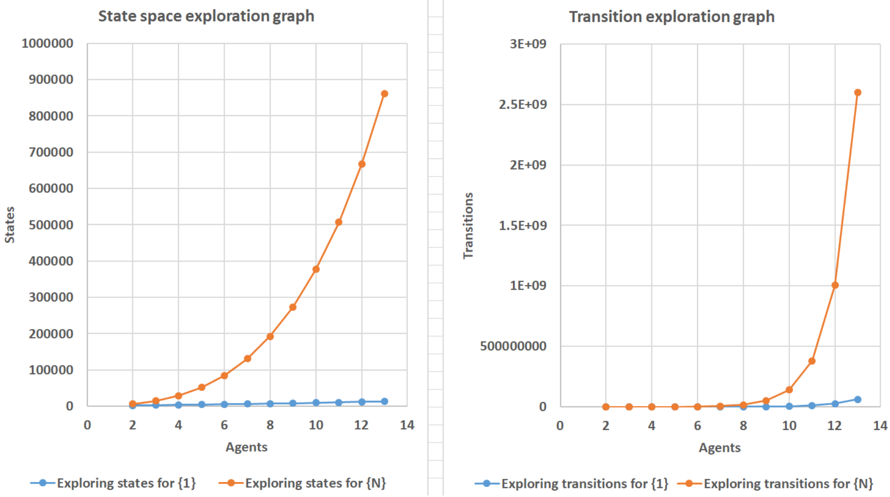

Fig. 7: State space and transition exploration graphs

problem become when resources are added explicitly. We have designed model checking algorithms for the resulting logic pRB-ATL, implemented them in a prototype tool, and used our techniques to solve simple multi-agent model checking problems of increasing complexity. Algorithm 1 returns a set of states satisfying a pRB-ATL formula φ and its complexity is $O(\mid \varphi \mid^{3r+1} \cdot \mid S \mid^3)$.

There are a number of interesting directions in this area for future research. First of all, we would like to investigate extensions of our logical framework and techniques to incorporate agent's behaviour with production of resources, and analyse a wider class of properties for the resulting logic. Secondly, we would like to study alternative semantics of the logics, including Interpreted Systems and Strategy Logic, implement them and report the expressivity and performance among alternative approaches. In this paper, we have used the example system just to explain the definitions/concepts used in Section 4 and in the rest of the paper in terms of the example. However, construction of a model considering a more realistic scenario is a non-trivial work. In future work, we have a plan to investigate the use of pRBATL logic for the analysis and verification of collaborative systems, by means of several use-cases. For example, in the domain of smart production system or similar other domains where a group of robots work collaboratively to achieve some goals.

References

[1] Alechina, N., Logan, B., Nguyen, H.N., Rakib, A.: Resource-bounded Alternating-time Temporal Logic. In: Proceedings of the 9th AAMAS: Volume 1, Toronto, Canada, pp. 481–488 (2010)

[2] Jennings, N.R., Wooldridge, M.: Applications of Intelligent Agents. In: Jennings, N.R., Wooldridge, M.J. (eds.) Agent Technology, pp. 3–28. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)

[3] Rapoport, A.: In: Eatwell, J., Milgate, M., Newman, P. (eds.) Prisoner's Dilemma, pp. 199–204. Palgrave Macmillan UK, London (1989)

[4] Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. J. ACM **49**(5), 672–713 (2002)

[5] Alur, R., de Alfaro, L., Grosu, R., Henzinger, T.A., Kang, M., Kirsch, C.M., Majumdar, R., Mang, F., Wang, B.Y.: jMocha: A model checking tool that exploits design structure. In: Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001, pp. 835–836. IEEE Comput. Soc, Toronto, Ont., Canada (2001)

[6] Pauly, M.: A Modal Logic for Coalitional Power in Games. Journal of Logic and Computation **12**(1), 149–166 (2002)

[7] Goranko, V.: Coalition Games and Alternating Temporal Logics. In: Proceedings of the 8th TARK, pp. 259–272. Morgan Kaufmann, Siena, Italy (2001)

[8] Ågotnes, T., van der Hoek, W., Wooldridge, M.: Reasoning about coalitional games. Artificial Intelligence **173**(1), 45–79 (2009)

[9] Herzig, A., Lorini, E., Walther, D.: Reasoning about actions meets strategic logics. In: Grossi, D., Roy, O., Huang, H. (eds.) Logic, Rationality, and Interaction, vol. 8196, pp. 162–175. Springer, Berlin, Heidelberg (2013)

[10] Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., Buttazzo, G.: The challenge of real-time multi-agent systems for enabling IoT and CPS. In: Proceedings of WI '17, pp. 356–364. ACM Press, Leipzig, Germany (2017)

[11] Leitao, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., Colombo, A.W.: Smart Agents in Industrial Cyber–Physical Systems. Proceedings of the IEEE **104**(5), 1086–1101 (2016)

[12] Faza, A.Z., Sedigh, S., McMillin, B.M.: Reliability analysis for the advanced electric power grid: From cyber control and communication to physical manifestations of failure. In: Buth, B., Rabe, G., Seyfarth, T. (eds.) Computer Safety, Reliability, and Security, vol. 5775, pp. 257–269. Springer, Berlin, Heidelberg (2009)

[13] Zhang, M., Selic, B., Ali, S., Yue, T., Okariz, O., Norgren, R.: Understanding uncertainty in cyber-physical systems: A conceptual model. In: Wasowski, A., Lönn, H. (eds.) Modelling Foundations and Applications, vol. 9764, pp. 247–264. Springer, Cham (2016)

[14] Abbas, W., Laszka, A., Vorobeychik, Y., Koutsoukos, X.: Scheduling Intrusion Detection Systems in Resource-Bounded Cyber-Physical Systems. In: Proceedings of the 1st CPS-SPC, pp. 55–66. ACM Press, Denver, Colorado, USA (2015)

[15] Laszka, A., Vorobeychik, Y., Koutsoukos, X.: Integrity assurance in resource-bounded systems through stochastic message authentication. In: Proceedings of HotSoS, pp. 1–12. ACM Press, Urbana, Illinois (2015)

[16] Kwiatkowska, M.: Advances and challenges of quantitative verification and synthesis for cyber-physical systems. In: SOSCYPS, pp. 1–5. IEEE, Vienna, Austria (2016)

[17] Kwiatkowska, M., Norman, G., Parker, D.: Prism: Probabilistic symbolic model checker. In: Field, T., Harrison, P.G., Bradley, J., Harder, U. (eds.) Computer Performance Evaluation: Modelling Techniques and Tools, vol. 2324, pp. 200–204. Springer, Berlin, Heidelberg (2002)

[18] Bianco, A., de Alfaro, L.: Model checking of probabilistic and nondeterministic systems. In: Thiagarajan, P.S. (ed.) Foundations of Software Technology and Theoretical Computer Science, vol. 1026, pp. 499–513. Springer, Berlin, Heidelberg (1995)

[19] Baier, C., Katoen, J.-P., Hermanns, H.: Approximative symbolic model checking of continuous-time markov chains. In: Baeten, J.C.M., Mauw, S. (eds.) CONCUR'99 Concurrency Theory, vol. 1664, pp. 146–161. Springer, Berlin, Heidelberg (1999)

[20] Baier, C.: On Algorithmic Verification Methods for Probabilistic Systems. Habilitation thesis, Fakultät für Mathematik & Informatik, Universität Mannheim (1998)

[21] Chen, T., Forejt, V., Kwiatkowska, M., Parker, D., Simaitis, A.: Automatic verification of competitive stochastic systems. Formal Methods in System Design **43**(1), 61–92 (2013)

[22] Nguyen, H.N., Rakib, A.: A probabilistic logic for resource-bounded multi-agent systems. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. IJCAI'19, pp. 521–527 (2019)

[23] Wooldridge, M., Dunne, P.E.: On the computational complexity of qualitative coalitional games. Arti. Intelligence **158**(1), 27–73 (2004)

[24] Goranko, V., Drimmelen, G.: Complete axiomatization and decidability of alternating-time temporal logic. Theoretical Computer Science **353**(1), 93–117 (2006)

[25] Bulling, N., Farwer, B.: Expressing properties of resource-bounded systems: The logics rtl* and rtl. In: Dix, J., Fisher, M., Novák, P. (eds.) Computational Logic in Multi-Agent Systems, vol. 6214, pp. 22–45. Springer, Berlin, Heidelberg (2010)

[26] Della Monica, D., Napoli, M., Parente, M.: On a Logic for Coalitional Games with Priced-Resource Agents. ENTCS **278**, 215–228 (2011)

[27] Chen, T., Lu, J.: Probabilistic Alternating-time Temporal Logic and Model Checking Algorithm. In: 4th FSKD, pp. 35–39. IEEE, Haikou, China (2007)

[28] Bulling, N., Jamroga, W.: What Agents Can Probably Enforce. Fundamenta Informaticae **93**(1-3), 81–96 (2009)

[29] Forejt, V., Kwiatkowska, M., Norman, G., Parker, D.: In: Bernardo, M., Issarny, V. (eds.) Automated Verification Techniques for Probabilistic Systems, pp. 53–113. Springer, Berlin, Heidelberg (2011)

[30] Huang, X., Su, K., Zhang, C.: Probabilistic Alternating-Time Temporal Logic of Incomplete Information and Synchronous Perfect Recall. In: Proceedings of the 26th AAAI, Toronto, Canada (2012)

[31] Song, F., Zhang, Y., Chen, T., Tang, Y., Xu, Z.: Probabilistic alternating-time

μ–calculus. In: AAAI, pp. 6179–6186 (2019)

[32] Fu, C., Turrini, A., Huang, X., Song, L., Feng, Y., Zhang, L.: Model checking probabilistic epistemic logic for probabilistic multiagent systems. In: Lang, J. (ed.) Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, pp. 4757–4763 (2018)

[33] Wan, W., Bentahar, J., Ben Hamza, A.: Model checking epistemic-probabilistic logic using probabilistic interpreted systems. Know.-Based Syst. **50**(C), 279–295 (2013)

[34] Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: Prism-games 3.0: Stochastic game verification with concurrency, equilibria and time. In: Lahiri, S.K., Wang, C. (eds.) Computer Aided Verification, pp. 475–487. Springer, Cham (2020)

[35] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, Moshe Vardi (eds.): Reasoning About Knowledge. MIT Press, Cambridge, Mass (1995)

[36] Naumov, P., Ros, K.: Strategic coalitions in stochastic games. Journal of Logic and Computation **31**(7), 1845–1867 (2021)

[37] Guan, J., Yu, N.: A probabilistic logic for verifying continuous-time markov chains. In: Fisman, D., Rosu, G. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, pp. 3–21. Springer, Cham (2022)

[38] Wang, Y., Zarei, M., Bonakdarpoor, B., Pajic, M.: Probabilistic conformance for cyber-physical systems. In: Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems. ICCPS '21, pp. 55–66. Association for Computing Machinery, New York, NY, USA (2021)

[39] Naumov, P., Ros, K.: Strategic coalitions in stochastic games. Journal of Logic and Computation, exab032, 2021 (2021)

[40] Broersen, J., Herzig, A., Troquard, N.: A normal simulation of coalition logic and an epistemic extension. In: Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge. TARK '07, pp. 92–101. Association for Computing Machinery, New York, NY, USA (2007)

[41] Smirnov., A., Sheremetov., L., Teslya., N.: Fuzzy Cooperative Games Usage in Smart Contracts for Dynamic Robot Coalition Formation: Approach and Use Case Description. In: Proceedings of the 21st International Conference on Enterprise Information Systems - Volume 1: ICEIS,, pp. 361–370 (2019)

[42] Billingsley, P.: Probability and Measure, 2nd edn. (1986)

[43] Baier, C., Katoen, J.-P.: Principles of Model Checking, (2008)

[44] Kwiatkowska, M., Norman, G., Parker, D.: Stochastic Model Checking. In: Bernardo, M., Hillston, J. (eds.) Formal Methods for Performance Evaluation, Springer vol. 4486, pp. 220–270 (2007)

[45] Raghavan, T.E.S., Filar, J.A.: Algorithms for stochastic games - A survey. ZOR Methods Model. Oper. Res. **35**(6), 437–472 (1991)

[46] Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd ed edn. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore (1996)

[47] ANTLR (ANother Tool for Language Recognition). https://www.antlr.org/. Accessed: 12-Mar-2020